

University of Groningen

Architectural assumptions and their management in software development

Yang, Chen

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2018

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Yang, C. (2018). *Architectural assumptions and their management in software development*. [Thesis fully internal (DIV), University of Groningen]. Rijksuniversiteit Groningen.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Bibliography

- [1] ISO. ISO/IEC/IEEE Std 42010-2011, Systems and Software Engineering – Architecture Description, 2011.
- [2] L. Bass, P. Clements, and R. Kazman. Software Architecture in Practice, 3rd edition, Addison-Wesley Professional, 2012.
- [3] P. Bourque and R.E. Fairley. Guide to the Software Engineering Body of Knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press, 2014.
- [4] R. Roeller, P. Lago, and H. van Vliet. Recovering architectural assumptions. *Journal of Systems and Software*, 79(4): 552-573, 2006.
- [5] C.B. Haley, R.C. Laney, J.D. Moffett, and B. Nuseibeh. Using trust assumptions with security requirements. *Requirements Engineering*, 11(2): 138-151, 2006.
- [6] M.M. Lehman and J.F. Ramil. Rules and tools for software evolution planning and management. *Annals of Software Engineering*, 11(1): 15-44, 2001.
- [7] G.A. Lewis, T. Mahatham, and L. Wrage. Assumptions Management in Software Development. Technical Report, CMU/SEI-2004-TN-021, 2004.
- [8] F.J. Corbató. On building systems that will fail. ACM Turing Award lectures, Year Awarded 1990, *Communications of the ACM*, 34(9): 72-81, 1991.
- [9] D. Garlan, R. Allen, and J. Ockerbloom. Architectural mismatch: Why reuse is still so hard. *IEEE Software*, 26(4): 66-69, 2009.
- [10] C. Yang, P. Liang, P. Avgeriou, U. Eliasson, R. Heldal, and P. Pelliccione. Architectural Assumptions and their Management in Industry – An Exploratory Study. In: *Proceedings of the 11th European Conference on Software Architecture (ECSA)*, Canterbury, UK, pp. 191-207, 2017.
- [11] A. Steingruebl and G. Peterson. Software assumptions lead to preventable errors. *IEEE Security & Privacy*, 7(4): 84-87, 2009.
- [12] A. Bazaz, J.D. Arthur, and J.G. Tront. Modeling security vulnerabilities: A constraints and assumptions perspective. In: *Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC)*, Indianapolis, IN, USA, pp. 95-102, 2006.
- [13] D. Van Landuyt and W. Joosen. Modularizing early architectural assumptions in scenario-based requirements. In: *Proceedings of the 17th International Conference on Fundamental Approaches to Software Engineering (FASE)*, Grenoble, France, pp. 170-184, 2014.
- [14] C. Yang, P. Liang, and P. Avgeriou. Assumptions and their management in software development: A systematic mapping study. *Information and Software Technology*, 94(2): 82-110, 2018.
- [15] R. Wieringa. Design science as nested problem solving. In: *Proceedings of the 4th International Conference on Design Science Research in Information*

- Systems and Technology (DESRIST), Philadelphia, Pennsylvania, USA, pp. 8–20, 2009.
- [16] F. Shull, J. Singer, and D.I.K. Sjøberg. Guide to Advanced Empirical Software Engineering. Springer, 2008.
 - [17] B. Kitchenham and S. Charters. Guidelines for Performing Systematic Literature Reviews in Software Engineering, Version 2.3. EBSE Technical Report EBSE-2007-01, Keele University and Durham University, 2007.
 - [18] A. Fink. The Survey Handbook, 2nd Edition. Sage. 2003.
 - [19] P. Runeson, M. Host, A. Rainer, and B. Regnell. Case Study Research in Software Engineering: Guidelines and Examples. John Wiley & Sons. 2012.
 - [20] B.G. Glaser and A.L. Strauss. The Discovery of Grounded Theory: Strategies for Qualitative Research. New York: Aldine Publishing, 1967.
 - [21] K. Beck et al. Manifesto for Agile Software Development. <http://www.agilemanifesto.org/>, accessed on 2017-08-09.
 - [22] P. Clements and M. Shaw, "The Golden Age of Software Architecture" Revisited. *IEEE Software*, 26(4): 70-72, 2009.
 - [23] R. Capilla, A. Jansen, A. Tang, P. Avgeriou, and M.A. Babar. 10 years of software architecture knowledge management: Practice and future. *Journal of Systems and Software*, 116(6): 191–205, 2015.
 - [24] J. Bosch. Software architecture: The next step. In *Proceedings of the 1st European Workshop on Software Architecture (EWSA)*, St Andrews, UK, pp. 194-199, 2004.
 - [25] P. Kruchten, P. Lago, and H. van Vliet. Building up and reasoning about architectural knowledge. In: *Proceedings of the 2nd International Conference on Quality of Software Architectures (QoSA)*, Västerås, Sweden, pp. 43-58, 2006.
 - [26] P. Clements, F. Bachmann, and L. Bass. Documenting Software Architectures: Views and Beyond, 2nd Edition. Addison-Wesley Professional, 2010.
 - [27] D. Van Landuyt, E. Truyen, and W. Joosen. Documenting early architectural assumptions in scenario-based requirements. In: *Proceeding of the Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and European Conference on Software Architecture (ECSA)*, Helsinki, Finland, pp. 329-333, 2012.
 - [28] R. Ali, F. Dalpiaz, P. Giorgini, and V.E.S. Souza. Requirements evolution: from assumptions to reality. In: *Proceedings of the 12th International Conference on Enterprise, Business-Process and Information Systems Modeling (BPMDs)*. London, UK, pp. 372-382, 2011.
 - [29] J. Arnowitz, M. Arent, and N. Berger. Verify Prototype Assumptions and Requirements. *User Experience Re-Mastered: Your Guide to Getting the Right Design*, Elsevier, pp. 221- 240, 2010.
 - [30] C. Blundell, D. Giannakopoulou, and C.S. Păsăreanu. Assume-guarantee testing. *ACM SIGSOFT Software Engineering Notes*, 31(2): Article No. 1,

- 2005.
- [31] C. Brenner, J. Greenyer, and V. Panzica La Manna. The ScenarioTools play-out of modal sequence diagram specifications with environment assumptions. *Electronic Communications of the EASST*, 58, 2013.
 - [32] J.M. Cobleigh, G.S. Avrunin, and L.A. Clarke. Breaking up is hard to do: An evaluation of automated assume-guarantee reasoning. *ACM Transactions on Software Engineering and Methodology*, 17(2): Article No. 7, 2007.
 - [33] V. De Florio. Software Assumptions Failure Tolerance: Role, Strategies, and Visions. *Architecting Dependable Systems VII*, Springer Berlin Heidelberg, pp. 249-272, 2010.
 - [34] P. Lago and H. van Vliet. Explicit assumptions enrich architectural models. In: *Proceedings of the 27th International Conference on Software Engineering (ICSE)*. St Louis, Missouri, USA, pp. 206-214, 2005.
 - [35] P. Kroll and P. Kruchten. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison-Wesley Professional, 2003.
 - [36] A.T. March and G.F. Smith. Design and natural science research on information technology. *Decision Support Systems*, 15(4): 251-266, 1995.
 - [37] A.R. Hevner, S.T. March, Park J, and S. Ram. Design science in information system research. *MIS Quarterly*, 28(1):75-105, 2004.
 - [38] R.J. Wieringa. *Design Science Methodology for Information Systems and Software Engineering*. Springer, 2014.
 - [39] V. Page, M. Dixon, and I. Choudhury. Security risk mitigation for information systems. *BT Technology Journal*, 25(1): 118-127, 2007.
 - [40] C. Flanagan, S.N. Freund, and S. Qadeer. Thread-modular verification for shared-memory programs. In: *Proceedings of the 11th European Symposium on Programming (ESOP)*, Grenoble, France, pp. 262-277, 2002.
 - [41] H. Ziv, D. Richardson, and R. Klösch. *The Uncertainty Principle in Software Engineering*. Technical Report, 1997.
 - [42] C. Wohlin, M. Höst, and K. Henningsson. Empirical Research Methods in Software Engineering. In: *Empirical Methods and Studies in Software Engineering*. Springer Berlin Heidelberg, pp. 7-23, 2003.
 - [43] M.S. Ali, M. Ali Babar, L. Chen, and K.J. Stol. A systematic review of comparative evidence of aspect-oriented programming. *Information and Software Technology*, 52(9): 871-887, 2010.
 - [44] L. Chen, M.A. Babar, and H. Zhang. Towards an evidence-based understanding of electronic data sources. In: *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering (ESEM)*, Bolzano-Bozen, Italy, pp. 135-138, 2010.
 - [45] V. Basili, G. Caldiera, and D. Rombach. *The Goal Question Metric Approach*. In: *Encyclopedia of Software Engineering*. John Wiley & Sons: New York, 1994.
 - [46] C. Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th International*

- Conference on Evaluation and Assessment in Software Engineering (EASE), London, UK, Article No. 38, 2014.
- [47] M.A.A. Mamun and J. Hansson. Review and challenges of assumptions in software development. In: Proceedings of the 2nd Analytic Virtual Integration of Cyber-Physical Systems Workshop (AVICPS), Vienna, Austria, Article No. 7, 2011.
 - [48] T. Dybå and T. Dingsøy. Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9): 833-859, 2008.
 - [49] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), Bari, Italy, pp.68-77, 2008.
 - [50] S. Adolph, W. Hall, and P. Kruchten. Using grounded theory to study the experience of software development. *Empirical Software Engineering*, 16(4): 487-513, 2011.
 - [51] P. Lago and H. van Vliet. Observations from the recovery of a software product family. In: Proceedings of the 3rd International Conference on Software Product Lines (SPLC), Boston, MA, USA, pp. 214-227, 2004.
 - [52] E. Denney and B. Fischer. A verification-driven approach to traceability and documentation for auto-generated mathematical software. In: Proceedings of the 24th IEEE/ACM International Conference on Automated Software Engineering (ASE), Auckland, New Zealand, pp. 560-564, 2009.
 - [53] R. Seater, D. Jackson, and R. Gheyi. Requirement progression in problem frames: Deriving specifications from requirements. *Requirements Engineering*, 12(2): 77-102, 2007.
 - [54] ARIANE 5 Flight 501 Failure Report by the Inquiry Board. 1996. <http://sunnyday.mit.edu/accidents/Ariane5accidentreport.html>
 - [55] D. Giannakopoulou, C.S. Păsăreanu, and H. Barringer. Component verification with automatically generated assumptions. *Automated Software Engineering*, 12(3): 297-320, 2005.
 - [56] S. Chaki, E. Clarke, N. Sharygina, and N. Sinha. Verification of evolving software via component substitutability analysis. *Formal Methods in System Design*, 32(3): 235-266, 2008.
 - [57] C. de la Riva and J. Tuya. Automatic generation of assumptions for modular verification of software specifications. *Journal of Systems and Software*, 79(9): 1324-1340, 2005.
 - [58] A. Tang, Y. Jin, and J. Han. A rationale-based architecture model for design traceability and reasoning. *Journal of Systems and Software*, 80(6): 918-934, 2007.
 - [59] J. Misra and K.M. Chandy. Proofs of networks of processes. *IEEE Transactions on Software Engineering*, 7(4): 417-426, 1981.

- [60] M. Abadi and L. Lamport. Conjoining specifications. *ACM Transactions on Programming Languages and Systems*, 17: 507-534, 1995.
- [61] C. Yang, P. Liang, and P. Avgeriou. Assumptions and Their Management in Software Development: A Systematic Mapping Study - Examples, Classifications, and Lessons Learned. http://www.cs.rug.nl/search/uploads/Resources/Assumptions_SMS-Examples_Classifications_and_Lessons_Learned.zip
- [62] B.A. Kitchenham and S.L. Pfleeger. Personal Opinion Surveys, in *Guide to Advanced Empirical Software Engineering*. Springer, pp. 63-92, 2008.
- [63] I. Ostacchini and M. Wermelinger. Managing assumptions during agile development. In: *Proceedings of the 4th Workshop on SHARing and reusing architectural Knowledge (SHARK)*, Vancouver, Canada, pp. 9-16, 2009.
- [64] A. Fink. *The Survey Handbook*. Sage Publications, Thousand Oaks, 1995.
- [65] T.C. Lethbridge. Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, 10(1): 311-341, 2005.
- [66] C. Wohlin, P. Runeson, M. Host, M.C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*, Springer, 2012.
- [67] W. Ding, P. Liang, A. Tang, H. van Vliet, and M. Shahin. How do open source communities document software architecture: An exploratory survey. In: *Proceedings of the 19th International Conference on Engineering of Complex Computer Systems (ICECCS)*, Tianjin, China, pp. 136-145, 2014.
- [68] A. Tang, M.A. Babar, I. Gorton, and J. Han. A survey of architecture design rationale. *Journal of Systems and Software*, 79(12): 1792-1804, 2006.
- [69] J. Shen and X. Yang. Analyzing urban competitiveness changes in major Chinese cities 1995-2008. *Applied Spatial Analysis and Policy*, 7(4): 361-379, 2014.
- [70] A. Tirumala, T. Crenshaw, L. Sha, G. Baliga, S. Kowshik, C. Robinson, and W. Witthawaskul. Prevention of failures due to assumptions made by software components in real-time systems. *ACM SIGBED Review*, 2(3): 36-39, 2005.
- [71] S. Zschaler and A. Rashid. Aspect assumptions: a retrospective study of AspectJ developers' assumptions about aspect usage. In: *Proceedings of the 10th International Conference on Aspect-Oriented Software Development (AOSD)*, Porto de Galinhas, Brazil, pp. 93-104, 2011.
- [72] S. Easterbrook, J. Singer, M.A. Storey, and D. Damian. Selecting empirical methods for software engineering research. In: *Guide to Advanced Empirical Software Engineering*, Springer London, pp. 285-311, 2008.
- [73] C. Larman. *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley Professional, 2003.
- [74] C. Larman and V.R. Basili. Iterative and incremental development: A brief history. *IEEE Computer*, 36(6): 47-56, 2003.
- [75] L. Del Greco and W. Walop. Questionnaire development: 1. Formulation. *Canadian Medical Association Journal*, 136(6): 583-585, 1987.

- [76] U. van Heesch and P. Avgeriou. Mature architecting - A survey about the reasoning process of professional architects. In: Proceedings of the 9th Working IEEE/IFIP Conference on Software Architecture (WICSA), Boulder, Colorado, USA, pp. 260-269, 2011.
- [77] K.A. de Graaf, P. Liang, A. Tang, and H. van Vliet. The impact of prior knowledge on searching in software documentation. In: Proceedings of the 6th ACM Symposium on Document Engineering (DocEng), Fort Collins, Colorado, USA, pp. 189-198, 2014.
- [78] J. Vanthienen and G. Wets. From decision tables to expert system shells. *Data & Knowledge Engineering*, 13(3): 265-282, 1994.
- [79] J. Corbin and A. Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 4th Edition. Sage Publications, 2014.
- [80] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2): 131-164, 2009.
- [81] B.W. Boehm. Software risk management: principles and practices. *IEEE Software*, 8(1): 32 - 41, 1991.
- [82] R.K. Yin. *Case study research. Design and methods*, 3rd edition. London, Sage, 2003.
- [83] A. Miransky, N. Madhavji, M. Davison, and M. Reesor. Modelling assumptions and requirements in the context of project risk. In: Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE), Paris, France, pp. 471 - 472, 2005.
- [84] S. Uchitel and D. Yankelevich. Enhancing architectural mismatch detection with assumptions. In: Proceedings of the 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), Edinburgh, UK, pp. 138-146, 2000.
- [85] H. Ordibehesht. Explicating critical assumptions in software architectures using AADL. Master Thesis, University of Gothenburg, 2010.
- [86] M.A.A. Mamun, M. Tichy, and J. Hansson. Towards Formalizing Assumptions on Architectural Level: A Proof-of-Concept. Research Report, University of Gothenburg, 2012.
- [87] M. William and J. Donnelly. *The Research Methods Knowledge Base*. Atomic Dog, 2006.
- [88] A. Tang, P. Avgeriou, A. Jansen, R. Capilla, and M. A. Babar. A comparative study of architecture knowledge management tools. *Journal of Systems and Software*, 83(3): 352-370, 2010.
- [89] W. Ding, P. Liang, A. Tang, and H. van Vliet. Knowledge-based approaches in software documentation: A systematic literature review. *Information and Software Technology*, 56(6): 545-567, 2014.
- [90] U. van Heesch, P. Avgeriou, and R. Hilliard. A documentation framework for architecture decisions. *Journal of Systems and Software*, 85(4): 795-820,

- 2012.
- [91] D. Rost, M. Naab, C. Lima, and C. von Flach Garcia Chavez. Software architecture documentation for developers: A survey. In: *Proceedings of the 7th European Conference (ECSA)*, Montpellier, France, pp. 72-88, 2013.
 - [92] C. Yang, P. Liang, P. Avgeriou, U. Eliasson, R. Heldal, P. Pelliccione, and T. Bi. Replication Package for AADF. http://www.cs.rug.nl/search/uploads/Resources/Replication_Package_for_AADF.zip
 - [93] T.M. Hesse and B. Paech. Supporting the collaborative development of requirements and architecture documentation. In: *Proceedings of the 3rd International Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks)*, Rio de Janeiro, Brazil, pp. 22-26, 2013.
 - [94] M. Höst, B. Regnell, and C. Wohlin. Using students as subjects - A comparative study of students and professionals in lead-time impact assessment. *Empirical Software Engineering*, 5(3): 201-214, 2000.
 - [95] U. van Heesch, P. Avgeriou, and A. Tang. Does decision documentation help junior designers rationalize their decisions? A comparative multiple-case study. *Journal of Systems and Software*, 86(6): 1545-1565, 2013.
 - [96] J.C. Carver, L. Jaccheri, S. Morasca, and F. Shull. A checklist for integrating student empirical studies with research and teaching goals. *Empirical Software Engineering*, 15 (1): 35-59, 2010.
 - [97] M. Kubickova and H. Ro. Are Students “Real People?” The Use of Student Subjects in Hospitality Research. http://scholarworks.umass.edu/gradconf_hospitality/2011/Poster/87/, 2011.
 - [98] P. Runeson. Using students as experiment subjects - An analysis on graduate and freshmen student data. In: *Proceedings of the 7th International Conference on Empirical Assessment in Software Engineering (EASE)*, Keele University, UK, pp. 95-102, 2003.
 - [99] P. Berander. Using students as subjects in requirements prioritization. In: *Proceedings of the 3rd International Symposium on Empirical Software Engineering (ISESE)*, Redondo Beach, CA, USA, pp. 167-176, 2004.
 - [100] D. Tofan, M. Galster, I. Lytra, P. Avgeriou, U. Zdun, M.A. Fouche, R. de Boer, and F. Solms. Empirical evaluation of a process to increase consensus in group architectural decision making. *Information and Software Technology*, 72(4): 31-47, 2016.
 - [101] M. Galster, D. Tofan, and P. Avgeriou. On integrating student empirical software engineering studies with research and teaching goals. In: *Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering (EASE)*, Ciudad Real, Spain, pp. 146-155, 2012.
 - [102] D.I.K. Sjøberg, J.E. Hannay, O. Hansen, V.B. Kampenes, A. Karahasanovic, N.K. Liborg, and A.C. Rekdal. A survey of controlled experiments in

- software engineering. *IEEE Transactions on Software Engineering*, 31(9): 733-753, 2005.
- [103] D.V. Landuyt and W. Joosen. On the role of early architectural assumptions in quality attribute scenarios: a qualitative and quantitative study. In: *Proceedings of the 5th International Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks)*, Florence, Italy, pp. 9-15, 2015.
- [104] J. Kontio, L. Lehtola, and J. Bragge. Using the focus group method in software engineering: obtaining practitioner and user experiences. In: *Proceedings of the 3rd International Symposium on Empirical Software Engineering (ISESE)*, Redondo Beach, CA, USA, pp. 271-280, 2004.
- [105] W. Nam, P. Madhusudan, and R. Alur. Automatic symbolic compositional verification by learning assumptions. *Formal Methods in System Design*, 32(3): 207-234, 2008.
- [106] A. Jansen. *Architectural Design Decisions*. PhD Thesis. University of Groningen, 2008.
- [107] R.C. de Boer, R. Farenhorst, P. Lago, H. van Vliet, V. Clerc, and A. Jansen. Architectural knowledge: Getting to the core. In: *Proceedings of the 3rd International Conference on Quality of Software Architectures (QoSA)*, Medford, MA, USA, pp. 197-214, 2007.
- [108] D. Turk, R. France, and B. Rumpe. Assumptions underlying agile software development processes. *Journal of Database Management*, 16(4): 62-87, 2005.
- [109] C. Hofmeister, P. Kruchten, R.L. Nord, H. Obbink, A. Ran, and P. America. A general model of software architecture design derived from five industrial approaches. *Journal of Systems and Software*, 80(1): 106-126, 2007.
- [110] U. van Heesch and P. Avgeriou. Naive architecting - Understanding the reasoning process of students A descriptive survey. In: *Proceedings of the 4th European Conference on Software Architecture (ECSA)*, Copenhagen, Denmark, pp. 24-37, 2010.
- [111] D. Tofan, M. Galster, and P. Avgeriou. Difficulty of architectural decisions - A survey with professional architects. In: *Proceedings of the 7th European Conference on Software Architecture (ECSA)*, Montpellier, France, pp. 192-199, 2013.
- [112] T.R. Knapp. Treating ordinal scales as interval scales: An attempt to resolve the controversy. *Nursing Research*, 39(2): 121-123, 1990.
- [113] H.N. Boone and D.A. Boone. Analyzing likert data. *Journal of Extension*, 50(2): 1-5, 2012.
- [114] A.O.S. Barreto and A.R. Rocha. Analyzing the similarity among software projects to improve software project monitoring processes. In: *Proceedings of the 7th International Conference on the Quality of Information and Communications Technology (QUATIC)*, Porto, Portugal, pp. 441-446, 2010.
- [115] R. Kazman, M. Klein, and P. Clements. *ATAM: Method for Architecture Evaluation*. Technical Report, 2000.
http://resources.sei.cmu.edu/asset_files/TechnicalReport/2000_005_001_1

- 3706.pdf
- [116] IEEE. IEEE Std 1517-2010, IEEE Standard for Information Technology - System and Software Life Cycle Processes - Reuse Processes, 2010.
 - [117] ISO. ISO/IEC/IEEE Std 14764-2006, Software Engineering - Software Life Cycle Processes - Maintenance, 2006.
 - [118] P. Abrahamsson, M. Babar, and P. Kruchten. Agility and architecture: Can they coexist?. *IEEE Software*, 27(2):16-22, 2010.
 - [119] M.A. Ali. Survey on the state of agile practices implementation in Pakistan. *International Journal of Information and Communication Technology Research*, 2(5):459-464, 2012.
 - [120] R.S. Arnold. *Software Change Impact Analysis*. IEEE Computer Society Press, 1996.
 - [121] S. Augustine. *Managing Agile Projects*. Prentice Hall: Upper Saddle River, NJ. 2005.
 - [122] M.A. Babar. An exploratory study of architectural practices and challenges in using agile software development approaches. In: *Proceedings of the Joint 8th Working IEEE/IFIP Conference on Software Architecture & 3rd European Conference on Software Architecture (WICSA/ECSA)*, Cambridge, UK, pp. 81-90, 2009.
 - [123] M.A. Babar, A.W. Brown, and I. Mistrik. *Agile Software Architecture: Aligning Agile Processes and Software Architectures*. Morgan Kaufmann. 2013.
 - [124] M.A. Babar, L. Zhu, and R. Jeffery. A framework for classifying and comparing software architecture evaluation methods. In: *Proceedings of the 15th Australian Software Engineering Conference (ASWEC)*, Melbourne, Australia, pp. 309-318, 2004.
 - [125] M.R. Barbacci, R. Ellison, A.J. Lattanze, J.A. Stafford, C.B. Weinstock, and W.G. Wood. *Quality Attribute Workshops (QAWs)*, Third Edition. Technical Report CMU/SEI-2003-TR-016, SEI, Carnegie Mellon University, 2003.
 - [126] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*, 2nd Edition. Addison-Wesley Professional, 2003.
 - [127] K. Beck et al. Principles behind the Agile Manifesto. <http://www.agilemanifesto.org/principles.html>, accessed on 2014-01-06.
 - [128] A. Begel and N. Nagappan. Usage and perceptions of agile software development in an industrial context: An exploratory study. In: *Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Madrid, Spain, pp. 255-264, 2007.
 - [129] P. Bengtsson, N. Lassing, J. Bosch, and H. van Vliet. Architecture-level modifiability analysis (ALMA). *Journal of Systems and Software*, 69(1):129-147, 2004.
 - [130] S. Blair, T. Cull, and R. Watt. Responsibility driven architecture. *IEEE Software*, 27(2):26-32, 2010.

- [131] B. Boehm, J.A. Lane, S. Koolmanojwong, and R. Turner. Architected Agile Solutions for Software-Reliant Systems, in *Agile Software Development*. Springer Berlin Heidelberg, pp. 165-184, 2010.
- [132] N. Brown, R. Nord, and I. Ozkaya. Enabling agility through architecture. *CrossTalk*, 13(6):12-17, 2010.
- [133] L. Cao, K. Mohan, and P. Xu. How extreme does extreme programming have to be? Adapting XP practices to large-scale projects. In: *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS)*, Big Island, Hawaii, USA, pp. 1-10, 2004.
- [134] L. Chen and M.A. Babar. Towards an evidence-based understanding of emergence of architecture through continuous refactoring in agile software development. In: *Proceedings of 11th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, Sydney, Australia, pp. 195-204, 2014.
- [135] H. Chivers, R.F. Paige, and X. Ge. Agile security using an incremental security architecture. In: *Proceedings of the 6th International Conference on Agile Processes in Software Engineering and Extreme Programming (XP)*, Sheffield, UK, pp. 57-65, 2005.
- [136] H.B. Christensen, K.M. Hansen, and B. Lindstrom. Lightweight and continuous architectural software quality assurance using the aSQA technique. In: *Proceedings of the 4th European Conference on Software Architecture (ECSA)*, Copenhagen, Denmark, pp. 118-132, 2010.
- [137] J. Diaz, J. Perez, and J. Garbajosa. Agile product-line architecting in practice: A case study in smart grids. *Information and Software Technology*, 56(7):727-748, 2014.
- [138] J. Diaz, J. Perez, J. Garbajosa, and A. Yague. Change-impact driven agile architecting. In: *Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS)*, Wailea, HI, USA, pp. 4780-4789, 2013.
- [139] T. Dingsøyr, T. Dybå, and N.B. Moe. *Agile Software Development: Current Research and Future Directions*. Springer, 2010.
- [140] L. Dobrica and E. Niemelä. A survey on software architecture analysis methods. *IEEE Transactions on Software Engineering*, 28(7): 638-653, 2002.
- [141] J. Erickson, K. Lyytinen, and K. Siau. Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management*, 16(4):88-100, 2005.
- [142] M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts. *Refactoring: Improving the Design of Existing Code*, Addison-Wesley Professional, 1999.
- [143] S. Freudenberg and H. Sharp. The top 10 burning research questions from practitioners. *IEEE Software*, 27(5):8-9, 2010.
- [144] I. Hadar, S. Sherman, E. Hadar, and J.J. Harrison. Less is more: Architecture documentation for agile development. In: *Proceedings of the 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, San Francisco, CA, USA, pp. 121-124, 2013.

-
- [145] A.M.M. Hamed and H. Abushama. Popular agile approaches in software development: Review and analysis. In: Proceedings of the International Conference on Computing, Electrical and Electronics Engineering (ICCEEE), Khartoum, Sudan, pp. 160-166, 2013.
 - [146] E. Hossain, M.A. Babar, and J. Verner. How can agile practices minimize global software development co-ordination risks? In: Proceedings of the 16th European Conference on Software Process Improvement (EuroSPI), Alcalá, Spain, pp.81-92, 2009.
 - [147] S. Jalali and C. Wohlin. Global software engineering and agile practices: A systematic review. *Journal of Software: Evolution and Process*, 24(6):643-659, 2012.
 - [148] R.N. Jensen, T. Moller, P. Sonder, and G. Tjornehoj. Architecture and design in extreme programming; introducing “developer stories”. In: Proceedings of the 7th International Conference on Agile Processes in Software Engineering and Extreme Programming (XP), Oulu, Finland, pp. 133-142, 2006.
 - [149] A. Josey, R. Harrison, P. Homan, M.F. Rouse, T. van Sante, M. Turner, and P. van der Merwe. *TOGAF Version 9.1 – A Pocket Guide*. Van Haren Publishing, 2011.
 - [150] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, and J. Carriere. The architecture tradeoff analysis method. In: Proceedings of the 4th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS), Monterey, CA, USA, pp. 68-78, 1998.
 - [151] P. Kruchten. The 4+1 view model of architecture. *IEEE Software*, 12(6): 42-50, 1995.
 - [152] P. Kruchten. Documentation of Software Architecture from a Knowledge Management Perspective – Design Representation, in *Software Architecture Knowledge Management*. Springer Berlin Heidelberg, pp. 39-57, 2009.
 - [153] P. Kruchten, R.L. Nord, and I. Ozkaya. Technical debt: From metaphor to theory and practice. *IEEE Software*, 29(6): 18-21, 2012.
 - [154] Z. Li, P. Liang, and P. Avgeriou. Application of knowledge-based approaches in software architecture: a systematic mapping study, *Information and Software Technology*. 55(5):777-794, 2013.
 - [155] I. Malavolta, P. Lago, H. Muccini, P. Pelliccione, and A. Tang. What industry needs from architectural languages: A survey. *IEEE Transactions on Software Engineering*, 39(6): 869-891, 2012.
 - [156] C. Manteuffel, D. Tofan, H. Koziol, T. Goldschmidt, and P. Avgeriou. Industrial implementation of a documentation framework for architectural decisions. In: Proceedings of the 11th Working IEEE/IFIP Conference on Software Architecture (WICSA), Sydney, Australia, pp. 225-234, 2014.
 - [157] N. Medvidovic and V. Jakobac. Using software evolution to focus architectural recovery. *Automated Software Engineering*, 13(2):225-256, 2006.

- [158] T. Mens, S. Demeyer, and T. Mens. Introduction and roadmap: History and challenges of software evolution, in *Software Evolution*. Springer Berlin Heidelberg, pp. 1-11, 2008.
- [159] R.L. Nord and J.E. Tomayko. Software architecture-centric methods and agile development. *IEEE Software*, 23(2):47-53, 2006.
- [160] B. Nuseibeh. Weaving together requirements and architecture. *IEEE Computer*, 34(3): 115-119, 2001.
- [161] K. Petersen and C. Wohlin. The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study. *Empirical Software Engineering*, 15(6):654-693, 2010.
- [162] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still. The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3):303-337, 2008.
- [163] A. Postma, P. America, and J.G. Wijnstra. Component replacement in a long-living architecture: the 3RDBA approach. In: *Proceedings of the 4th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, Oslo, Norway, pp. 89-98, 2004.
- [164] C.R. Prause and Z. Durdik. Architectural design and documentation: Waste in agile development?. In: *Proceedings of the International Conference on Software and System Process (ICSSP)*, Zurich, Switzerland, pp. 130-134, 2012.
- [165] J. Ramil and M. Lehman, Metrics of software evolution as effort predictors - a case study. In: *Proceedings of the 16th International Conference on Software Maintenance (ICSM)*, San Jose, CA, USA, pp. 163-172, 2000.
- [166] P. Rodríguez, J. Markkula, M. Oivo, and K. Turula. Survey on agile and lean usage in finnish software industry. In: *Proceedings of the 6th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Lund, Sweden, pp. 139-148, 2012.
- [167] K. Schwaber. Scrum development process. In: *Proceedings of the 10th Annual Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA) Workshop on Business Object Design and Implementation*, Austin, Texas, USA, pp. 117-134, 1995.
- [168] K. Schwaber and M. Beedle. *Agile Software Development with Scrum*. Prentice Hall PTR Upper Saddle River, 2002.
- [169] P. Sfetsos and I. Stamelos. Empirical studies on quality in agile practices: A systematic literature review. In: *Proceedings of the 7th International Conference on the Quality of Information and Communications Technology (QUATIC)*, Porto, Portugal, pp. 44-53, 2010.
- [170] M. Shahin, P. Liang, and M.A. Babar. A systematic review of software architecture visualization techniques. *Journal of Systems and Software*, 94(8): 161-185, 2014.
- [171] A.A. Sharifloo, A.S. Saffarian, and F. Shams. Embedding architectural practices into extreme programming. In: *Proceedings of the 19th Australian*

- Conference on Software Engineering (ASWEC), Perth, WA, Australia, pp. 310-319, 2008.
- [172] F.S. Silva, F.S.F. Soares, A.L. Peres, I.M. de Azevedo, A.P.L. Vasconcelos, F.K. Kamei, and S.R. de Lemos Meira. Using CMMI together with agile software development: A systematic review. *Information and Software Technology*, 58(2):20-43, 2014.
 - [173] J. Stapleton. *Dynamic Systems Development Method: The Method in Practice*. Addison-Wesley: Reading, 1998.
 - [174] S. Stavru. A critical examination of recent industrial surveys on agile method usage. *Journal of Systems and Software*, 94(8):87-97, 2014.
 - [175] A.L. Strauss. *Qualitative Analysis for Social Scientists*. Cambridge University Press, 1987.
 - [176] M.T. Su. Capturing exploration to improve software architecture documentation. In: *Proceedings of the 4th European Conference on SoftwareArchitecture (ECSA)*, Copenhagen, Denmark, pp. 17-21, 2010.
 - [177] P. Thapparambil. Agile architecture: pattern or oxymoron? *Agile Times*, 6(1): 43-48, 2005.
 - [178] U. van Heesch, V.P. Eloranta, P. Avgeriou, K. Koskimies, and N. Harrison. DCAR - Decision-Centric Architecture Reviews. *IEEE Software*, 31(1):69-76, 2014.
 - [179] VersionOne. The 9th annual state of agile survey. <http://info.versionone.com/state-of-agile-development-survey-ninth.html>, 2015.
 - [180] I. Rus, and M. Lindvall. Knowledge management in software engineering. *IEEE Software*, 19(3):26-38, 2002.
 - [181] J. Tyree, and A. Akerman. Architecture decisions: Demystifying architecture. *IEEE Software*, 22(2):19-27, 2005.
 - [182] M. Shahin, P. Liang, and M. R. Khayyambashi. Architectural design decision: Existing models and tools. In *Proceedings of the Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture (WICSA/ECSA)*, pp. 293-296, 2009.
 - [183] P. Kruchten. An ontology of architectural design decisions in software intensive systems. In: *Proceedings of the 2nd Groningen Workshop on Software Variability Management (SVM)*, pp. 54-61, 2004.
 - [184] F.D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3): 319-340, 1989.
 - [185] V. Venkatesh and F.D. Davis. A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, 46(2): 186 - 204, 2000.
 - [186] V. Venkatesh and H. Bala. Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences*, 39(2): 273-315, 2008.
 - [187] C. Manteuffel, D. Tofan, P. Avgeriou, H. Kozirolek, and T. Goldschmidt. Decision architect-A decision documentation tool for industry. *Journal of*

- Systems and Software, 112(2): 181–198, 2016.
- [188] T. Heyman, R. Scandariato, and W. Joosen. Security in context: analysis and refinement of software architectures. In: Proceedings of the 34th Annual IEEE Computer Software and Applications Conference (COMPSAC). Seoul, South Korea: pp. 161-170, 2010.
 - [189] K. Welsh, P. Sawyer, and N. Bencomo. Towards requirements aware systems: Run-time resolution of design-time assumptions. In: Proceedings of the 26th IEEE/ACM International Conference on Automated Software Engineering (ASE). Lawrence, KS, USA: pp. 560-563, 2011.
 - [190] S. Faily and I. Fléchaïs. The secret lives of assumptions: Developing and refining assumption personas for secure system design. In: Proceedings of the 3rd International Conference on Human-Centred Software Engineering (HCSE). Reykjavik, Iceland: pp. 111-118, 2010.
 - [191] I. Habli and T. Kelly. Capturing and replaying architectural knowledge through derivational analogy. In: Proceedings of the 2nd Workshop on SHARing and Reusing architectural Knowledge Architecture, Rationale, and Design Intent (SHARK-ADI). Minneapolis, MN, USA: Article No. 4, 2007.
 - [192] ArAM, Its Description, and Replication Package for the Case Study. http://www.cs.rug.nl/search/uploads/Resources/ArAM_Its%20Description_and%20Replication%20Package%20for%20the%20Case%20Study.zip
 - [193] M. Alavi and D.E. Leidner. Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Quarterly*, 25(1): 107-136, 2001.
 - [194] A. Aurum, R. Jeffery, C. Wohlin, and M. Handzic. *Managing Software Engineering Knowledge*, Springer-Verlag, New York, 2003.
 - [195] H. Ordibehesht. *Explicating Critical Assumptions in Software Architectures Using AADL*. University of Gothenburg, 2010.